KERJA PRAKTIK – IF184801

Pembuatan Fitur Manage Member dan Login Pada Aplikasi CCTools untuk Kebutuhan Sistem Informasi Karyawan Pada Bidang Customer Care

PT. Beon Intermedia
Jalan MT Haryono No. 1 Blok C4, Malang
Periode: **6 Juli 2020 – 6 Oktober 2020**

Oleh:

Pembimbing Jurusan
Prof. Ir. Supeno Djanali, M.Sc., Ph.D.

Pembimbing Lapangan
Adi Susanto

DEPARTEMEN INFORMATIKA
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember
Surabaya 2020

KERJA PRAKTIK – IF184801

Pembuatan Fitur Manage Member dan Login Pada Aplikasi CCTools untuk Kebutuhan Sistem Informasi Karyawan Pada Bidang Customer Care

PT. Beon Intermedia
Jalan MT Haryono No. 1 Blok C4, Malang
Periode: 6 Juli 2020 – 6 Oktober 2020

Oleh:

Pembimbing Jurusan
Prof. Ir. Supeno Djanali, M.Sc., Ph.D.

Pembimbing Lapangan
Adi Susanto

DEPARTEMEN INFORMATIKA
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember
Surabaya 2020

[Halaman ini sengaja dikosongkan]

LEMBAR PENGESAHAN

KERJA PRAKTIK

Pembuatan Fitur Manage Member dan Login Pada Aplikasi CCTools untuk Kebutuhan Sistem Informasi Karyawan Pada Bidang Customer Care

Oleh:

Ahmad Yahya Abdul Aziz 05111740000128

Menyetujui,

Dosen Pembimbing Kerja Praktik

(Prof. Ir. Supeno Djanali, M.Sc., Ph.D.)
NIP. 194806191973011001

SURABAYA

6 OKTOBER, 2020

[Halaman ini sengaja dikosongkan]

Pembuatan Fitur Manage Member dan Login Pada Aplikasi CCTools untuk Kebutuhan Sistem Informasi Karyawan Pada Bidang Customer Care

Nama Mahasiswa	**: Ahmad Yahya Abdul Aziz**
NRP	**: 05111740000128**
Departemen	**: Informatika FTEIC-ITS**
Pembimbing Jurusan	**: Supeno Djanali**
Pembimbing Lapangan	**: Adi Susanto**

[Halaman ini sengaja dikosongkan]

ABSTRAK

PT. Beon Intermedia merupakan sebuah perusahaan yang telah melayani lebih dari 7000 customer hosting Indonesia. Sebagai perusahaan yang sudah berjalan selama 8 tahun di bidang jasa layanan hosting, berbagai produk ditawarkan seperti produk jasa layanan cloud hosting hingga jasa vps yang murah dan cepat. Dengan semakin berkembangnya perusahaan ini dan jumlah karyawan yang terbatas, dibutuhkan sebuah sistem informasi yang berguna membantu pekerjaan karyawan Beon di bidang Customer Care dalam melayani para customer.

Untuk memaksimalkan kinerja dari para karyawan Beon bidang Customer Care, dibutuhkan sebuah aplikasi dengan fitur login dan manage member yang bertujuan untuk memonitoring customer yang berguna untuk memastikan bahwa proses pemesanan customer untuk sebuah produk lancar sampai dapat digunakan oleh customer tersebut. Dengan adanya sistem login, maka hanya karyawan yang dapat mengakses, sehingga data customer terjaga.

Dengan menggunakan Bahasa pemrograman web dan database seperti HTML, CSS, JS, PHP, SQL, dan tools pendukung seperti Visual Studio Code serta Framework seperti NodeJs, VueJs, dan Phalcon yang bertujuan untuk menunjang pembuatan aplikasi tersebut. Dengan adanya aplikasi tersebut, diharapkan dapat mempermudah karyawan di bidang Customer Care untuk memonitoring para customer apabila terjadi masalah sehingga memenuhi target dari perusahaan.

Kata kunci: Monitoring, login, member, framework

KATA PENGANTAR

Puji syukur saya haturkan kepada Allah SWT karena berkat rahmat-Nya saya dapat melaksanakan salah satu kewajiban saya sebagai mahasiswa Departemen Informatika ITS, yakni Kerja Praktek (KP).

Saya menyadari masih ada banyak kekurangan baik dalam pelaksanaan kerja praktik maupun penyusunan buku laporan ini. Namun, saya berharap buku laporan ini dapat menambah wawasan pembaca dan juga dapat menjadi sumber referensi bagi orang yang membacanya. Saya mengharapkan kritik dan saran yang membangun untuk membantu kesempurnaan buku laporan kerja praktik ini.

Melalui buku ini, saya juga ingin menyampaikan rasa terima kasih kepada orang-orang yang telah banyak membantu, baik secara langsung maupun tidak langsung selama pelaksanaan kerja praktik hingga penyusunan laporan ini. Orang-orang tersebut antara lain adalah:

1. Orang tua penulis.
2. Bapak Ary Mazharuddin Shiddiqi, S. Kom., M. Comp., Ph. D. selaku koordinator kerja praktik.
3. Bapak Prof. Ir. Supeno Djanali, M. Sc., Ph. D., selaku dosen pembimbing kerja praktik.
4. Bapak Adi Susanto, selaku pembimbing lapangan PT. Beon Intermedia divisi Operational Management
5. Bapak Andhika Martha Wijaya, selaku salah satu karyawan PT. Beon Intermedia divisi Operational Management

Surabaya, Oktober 2020

Ahmad Yahya Abdul Aziz

[Halaman ini sengaja dikosongkan]

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

DAFTAR KODE

BAB I
PENDAHULUAN

1.1. Latar Belakang

Seiring berjalannya waktu, semakin banyak bisnis startup yang terdapat di Indonesia. Semakin banyak orang-orang yang berminat untuk mencoba mengembangkan bisnis mereka ke arah digital. Untuk dapat mengembangkan bisnis ke arah digital, orang-orang akan membutuhkan hosting untuk membuat website mereka. Hal ini membuat peminat layanan hosting semakin banyak. PT. Beon Intermedia yang memiliki bisnis ke arah layanan hosting juga akan semakin banyak customer nya. Hal tersebut menjadikan perusahaan ini membutuhkan sebuah sistem yang dapat memudahkan dalam mengatur dan memonitoring customer. Oleh karena itu dibuatlah aplikasi CCTools ini.

Aplikasi CCTools ini berfungsi sebagai aplikasi yang dapat memudahkan karyawan dalam mengecek apabila terdapat masalah saat customer melakukan pembayaran. Tidak hanya itu, aplikasi ini juga berguna untuk mengingatkan para customer yang belum menyelesaikan pembayaran. Dikarenakan terbatasnya sumber daya manusia di bidang IT yang memiliki kompetensi dalam mengembangkan aplikasi ini, PT. Beon Intermedia membutuhkan orang untuk membantu mengembangkan aplikasi ini.

1.2. Tujuan

Tujuan kerja praktik ini adalah untuk menyelesaikan kewajiban kuliah kerja praktik di Institut Teknologi Sepuluh Nopember dengan beban dua SKS. Selain itu kerja praktik ini juga bertujuan membantu PT. Beon Intermedia untuk membuat aplikasi yang dapat memudahkan karyawan untuk monitoring customer.

1.3. Manfaat

Sama seperti tujuan di atas, manfaat dari pembuatan aplikasi CCTools adalah untuk memudahkan karyawan dalam monitoring customer.

1.4. Rumusan Masalah

Berikut ini rumusan masalah pada kerja praktik pembuatan fitur login dan manage member pada aplikasi CCTools di PT. Beon Intermedia:

1. Bagaimana pegawai PT. Beon Intermedia bagian Customer Care dapat akses aplikasi CCTools secara aman?
2. Bagaimana pegawai PT. Beon Intermedia bagian Customer Care mengatur akses pengguna aplikasi CCTools?
3. Bagaimana pegawai PT. Beon Intermedia bagian Customer Care menambahkan pengguna yang dapat akses aplikasi CCTools?

1.5. Lokasi dan Waktu Kerja Praktik

Kerja praktik ini dilaksanakan pada waktu dan tempat sebagai berikut:

Lokasi	: PT. Beon Intermedia
Alamat	: Jalan MT Haryono No. 1 Blok C4, Malang
Waktu	: 6 Juli 2020 – 6 Oktober 2020
Hari Kerja	: Senin – Jumat
Jam Kerja	: 08.00 WIB – 17.00 WIB

1.6. Metodologi Kerja Praktik

Tahapan pengerjaan kerja praktik dapat dijabarkan sebagai berikut:

1. Perumusan Masalah

Untuk mengetahui permasalahan apa yang harus diselesaikan, diberikan penjelasan mengenai apa saja yang dibutuhkan dari aplikasi. Dijelaskan juga secara detail mengenai alur sistem yang akan berjalan. Penjelasan mengenai hal – hal ini dijelaskan oleh pegawai bagian Customer Care kepada penulis dan pembimbing lapangan kerja praktik, yang kemudian penulis dan pembimbing lapangan kerja praktik melakukan diskusi yang

menghasilkan catatan-catatan penting mengenai gambaran sistem yang akan dibuat dan fitur yang berguna untuk mempermudah kinerja Customer Care.

Dari hal tersebut diputuskan bahwa system yang akan dibuat akan berbasis web, dengan Bahasa pemrograman yang digunakan adalah HTML, CSS, JSS, PHP, dan MySQL, dengan memanfaatkan framework yang ada seperti VueJs sebagai pondasi dari frontend nya dan Phalcon sebagai pondasi dari backend nya. Dari situ, penulis menggunakan tools pendukung seperti Visual Studio Code, dan NodeJs untuk dapat menggunakan framework tersebut.

2. Studi Literatur

Setelah menentukan apa saja yang akan dibutuhkan dalam pembuatan system tersebut, dilakukan studi literatur mengenai cara implementasi dalam menggunakan framework yang ada. Pada tahap ini, dilakukan proses pencarian, pembelajaran, dan pemahaman informasi dari literatur yang berkaitan. Informasi dari literatur bisa didapat dari berbagai dokumentasi yang terdapat di Internet, sehingga semakin mempermudah penulis dalam melakukan tahap ini.

3. Analisis dan Perancangan

Pada tahap ini penulis melakukan analisis mengenai bagaimana fitur akan dibuat berdasarkan studi literatur yang telah dilakukan. Tahap ini terbagi menjadi dua, yaitu tahap analisis dan perancangan frontend, dan backend.

Langkah-langkah yang dikerjakan pada tahap perancangan backend dapat didefinisikan sebagai berikut:

a. Memahami elemen dan isi dari seluruh tabel yang ada di dalam database
b. Memahami primary key dan foreign key yang ada pada tabel

c. Menggunakan Visual Studio Code sebagai tools pendukung yang digunakan dalam membuat fungsi query pada Phalcon (menggunakan PHP dan MySQL).

Sedangkan langkah-langkah yang dikerjakan pada tahap perancangan frontend adalah sebagai berikut:

a. Memahami desain tampilan yang akan dibuat pada halaman member.
b. Memahami bentuk framework VueJs dalam membuat halaman tersebut.
c. Menggunakan Visual Studio Code sebagai tools pendukung yang digunakan dalam membuat halaman member pada VueJS (menggunakan HTML, CSS, JS).

4. Implementasi Sistem

Implementasi sistem didasarkan kepada adanya kebutuhan dari fullstack developer untuk membuat suatu sistem berbasis web untuk mengecek status pesanan dari customer. Pada tahap ini dibuat berbagai fungsi query di backend oleh developer sesuai kebutuhan dari karyawan divisi Customer Care. Pada tahap ini juga dibuat tampilan frontend setiap fitur yang ada tergantung dari permintaan dan kebutuhan. Pengerjaan dilakukan selama kurang lebih dua bulan / delapan minggu.

5. Pengujian dan Evaluasi

Pengujian dilakukan dengan melakukan pengecekan setiap fungsi pada tiap halaman yang dilakukan secara otomatis menggunakan salah satu framework JavaScript yaitu Cypress. Selain dicek secara otomatis, terdapat dokumentasi UAT (User Acceptance Testing) yang dibuat developer untuk dites oleh pengguna (karyawan Customer Care) apakah sudah sesuai kebutuhan. Program automatic testing dengan Cypress dikerjakan selama kurang lebih satu bulan / empat minggu.

6. Kesimpulan dan Saran

Kesimpulan yang didapatkan selama kerja praktik adalah perlunya pemahaman dalam menterjemahkan kebutuhan yang telah dijelaskan oleh pengguna. Selain itu, diperlukan komunikasi secara berkala antara developer dan pengguna mengenai progress pembuatan sistem apakah sudah memenuhi kebutuhan pengguna, agar tidak terjadi kesalahan dalam membuat sistem.

1.7. Sistematika Laporan

Laporan kerja praktik ini terdiri dari enam bab dengan rincian sebagai berikut:

1. Bab I Pendahuluan

Pada bab ini dijelaskan tentang latar belakang permasalahan, tujuan, waktu pelaksanaan, serta sistematika pengerjaan kerja praktik dan juga penulisan laporan kerja praktik.

2. Bab II Profil Perusahaan

Pada bab ini, dijelaskan secara rinci tentang profil perusahaan tempat dilaksanakannya kerja praktik, yakni PT. Beon Intermedia

3. Bab III Tinjauan Pustaka

Pada bab ini, dijelaskan mengenai tinjauan pustaka dan literatur yang digunakan dalam penyelesaian kerja praktik di PT. Beon Intermedia.

4. Bab IV Implementasi Sistem

Pada bab ini, berisi penjelasan tahap-tahap yang dilakukan untuk proses implementasi fungsi query dalam backend menggunakan framework Palchon dan tampilan dalam frontend menggunakan framework VueJS.

5. Bab V Pengujian dan Evaluasi

Pada bab ini, dijelaskan tentang hasil pengujian dan evaluasi dari sistem yang telah dikembangkan selama pelaksanaan kerja praktik di PT. Beon Intermedia.

6. Bab VI Kesimpulan dan Saran

Pada bab ini, dipaparkan kesimpulan yang dapat diambil dan juga saran selama pengerjaan kerja praktik.

BAB II
PROFIL PERUSAHAAN

2.1. Profil PT. Beon Intermedia

Gambar 2.1 Logo PT. Beon Intermedia

PT. Beon Intermedia adalah perusahaan yang bergerak pada penyedia jasa layanan sewa cloud hosting dan virtual private server termurah di Indonesia. Beon mengklaim sebagai salah satu perusahaan layanan cloud hosting besar di Indonesia yang telah merambat hingga ke luar negeri juga dengan jumlah pelanggan lebih dari 7000.

Beon merupakan perusahaan jasa layanan cloud hosting yang telah berdiri dari tahun 2007, dengan berbagai penghargaan yang telah diraih, diantaranya adalah Wirausaha Muda Mandiri Jawa Timur pada tahun 2009, ISMBEA (Indonesia Small Medium Business Entrepreneur Award) pada tahun 2009, dan Sparxup Awards pada tahun 2011 sebagai best E-commerce.

2.2. Visi dan Misi PT. Beon Intermedia

PT. Beon Intermedia memiliki visi dan misi yang terus menjadi patokan perusahaan untuk berkembang, berikut adalah visi dan misi PT. Beon Intermedia

- **Visi** : Memberi manfaat lebih dari yang anda bayangkan
- **Misi** : Menyediakan berbagai layanan yang unggul dan profesional bagi customer dan potensial customer.

Dengan visi tersebut, Beon menjelaskan bahwa mereka akan selalu memberikan yang terbaik bagi customer. Terbaik dalam layanan maupun layanan. Visi tersebut akan diwujudkan melalui misi yang maksudnya adalah Beon akan terus mengembangkan diri untuk menjadi yang terbaik dan terus berkompetisi dengan diri sendiri untuk menjadi lebih baik, agar selalu melayani customer dengan baik dan lebih baik lagi.

2.3. Produk PT. Beon Intermedia

Produk dari PT. Beon Intermedia diantaranya adalah sebagai berikut:

- Cloud Hosting
  - ID FAME
  - ID HITS
  - ID HITS-DEVELOPER
  - ID SUPERSTAR
  - ID SUPERSTAR-DEVELOPER
  - SG FAME
  - SG HITS
  - SG SUPERSTAR
- Corporate Hosting
  - B2B
    - UNI
    - DECA
    - HECTO
  - B2C
    - ANGEL
    - SEED
    - VENTURE
- Cloud Mail
- Cloud Mail Bundling Domain

- Mail Hosting
  - Giga
  - Tera
  - Peta
  - Espresso
  - Espresso Addict
- Jagoan SSL
  - SSL: Comodo Positive Wildcard
  - SSL: Comodo Positive Green Bar
  - SSL: Comodo Positive Single Domain
- Unlimited Hosting
  - Beon Unlimited JAVA
  - Beon Unlimited GAYO
  - Beon Unlimited SUMATRA
  - Beon US Unlimited JAVA
  - Beon US Unlimited GAYO
  - Beon US Unlimited SUMATRA
- Reseller Cpanel
  - Cpanel-Micro
  - Cpanel-Mini
  - Cpanel-Medium
  - Cpanel-Big
  - Cpanel-Giant
- License Cpanel
  - Cpanel License-Solo Cloud
  - Cpanel License-Admin Cloud
  - Cpanel License-Pro Cloud
  - Cpanel License-Plus Cloud
  - Cpanel License-Premiere Cloud 100
  - Cpanel License-Premiere Cloud 200

- VPS X (Virtual Private Server)
  - X1 (disk : 40GB)
  - X2 (disk : 80GB)
  - X3 (disk : 120GB)
- Dedicated Hosting
  - ARAMON
  - BARBERA
  - CORVINA
  - DOMINA
- VM X (Virtual Machine)
  - X PRO LITE (disk: 20GB)
  - X PRO PLUS (disk: 40GB)
  - X PRO MAX (disk: 60GB)

BAB III
TINJAUAN PUSTAKA

Pada bab ini, akan dijelaskan mengenai dasar teori yang digunakan selama proses kerja praktik.

3.1. HTML

Hyper Text Markup Language atau yang biasa disingkat dengan HTML merupakan standard *markup language* yang digunakan untuk menampilkan dokumen dalam sebuah website. HTML ada bertujuan untuk membuat dokumen terstruktur seperti text yang terbagi atas heading, paragraphs, list, dan bagian lainnya. Setiap struktur dari HTML terdiri atas label yang secara langsung menjelaskan isi dari label ke halaman tersebut. Browser sendiri berguna untuk menginterpretasikan label tersebut menjadi sebuah konten dalam sebuah halaman.

Dalam membuat sebuah website, HTML merupakan dasar dari website, dimana terdapat teknologi lainnya yang dapat membantu mengubah tampilan, atau fungsi dari beberapa bagian dari halaman seperti CSS, dan JavaScript.

3.2. JavaScript (JS)

JavaScript merupakan salah satu *scripting language* yang paling banyak digunakan developer dalam membuat website. Sebagai salah satu inti dari pembuatan website, JavaScript memungkinkan sebuah website yang interaktif dan merupakan bagian penting dari sebuah web aplikasi. Sebagai sebuah *multi-paradigm language*, JavaScript mendukung fungsional, event-handler, dan lainnya yang berguna untuk menjadikan sebuah website interaktif.

Sebagian besar website menggunakan JavaScript untuk membuat *client-side* website. Selama perkembangannya, JavaScript memungkinkan digunakan tidak hanya website saja, namun hingga ke server juga. Hal ini dikarenakan terdapat NodeJS dengan

berbagai framework berbasis JavaScript di dalamnya yang membantu memudahkan dalam pemrograman website.

3.3. MySQL

Gambar 3.1 Logo MySQL

MySQL merupakan sebuah open-source *Relational Database Management System* (RDBMS). Data disimpan dalam berbagai format. MySQL merupakan salah satu sistem database untuk mengelola data yang membantu dalam menyusun kumpulan data. SQL sendiri merupakan pemrograman yang berguna untuk membuat, mengubah, dan mengambil data dari database yang ada.

3.4. Visual Studio Code

Gambar 3.2 Logo Visual Studio Code

Visual Studio Code merupakan code editor yang mendukung banyak bahasa pemrograman dan bahasa *markup*. Dengan memanfaat extension yang ada, developer akan lebih mudah dalam membuat program. Fitur yang diberikan oleh visual studio code memudahkan developer untuk membuat aplikasi, dengan terminal untuk menjalankan program, snippets untuk mempercepat dalam mengetik program, dan berbagai extension yang semakin memudahkan untuk membuat dan melakukan testing terhadap sebuah program.

3.5. NodeJS

Gambar 3.3 Logo NodeJS

NodeJS merupakan sebuah JavaScript runtime yang didesain untuk membangun jaringan aplikasi. Berbeda dengan sistem sejenis lainnya yang perlu sebuah blok untuk menjalankan sebuah event-loop, NodeJS akan masuk ke event-loop langsung setelah menjalankan input script. Tindakan ini seperti halnya browser JavaScript, dimana event-loop tidak terlihat oleh pengguna.

HTTP adalah hal penting di dalam NodeJS, didesain dengan dasar streaming dan low latency. Hal ini menjadikan NodeJS sangat cocok untuk digunakan sebagai fondasi dari sebuah web library atau framework.

3.6. VueJS

Gambar 3.4 Logo VueJS

VueJS adalah sebuah *framework* yang digunakan untuk membangun *user interface*. Berbeda dengan *framework* lainnya, Vue didesain terfokus pada *view layer*, yaitu tampilan dari sebuah halaman website, dan sangat mudah untuk diintegrasikan dengan library lainnya ataupun suatu project yang ada. VueJS juga mampu

dalam membuat *single-page application* saat dikombinasikan dengan teknologi modern dan *library* lain yang mendukung.

3.7. Phalcon

Gambar 3.5 Logo Phalcon

Phalcon merupakan fullstack framework berbasis PHP yang sudah dioptimalkan untuk *high performance*. Framework ini memungkinkan untuk selalu menghabiskan sedikit memori, menggunakan fungsinya setiap dibutuhkan tanpa file stats dan file reads yang berat seperti PHP framework lainnya.

Phalcon merupakan framework yang jenisnya *loosely coupled*, memungkinkan untuk developer menggunakan objek yang digunakan saja sesuai kebutuhan aplikasinya.

3.8. Cypress

Gambar 3.6 Logo Cypress

Cypress adalah sebuah framework berbasis JavaScript yang digunakan untuk melakukan testing fungsi dari suatu website secara otomatis. Berbeda dengan teknologi testing lainnya yang berbasis Selenium. Cypress merupakan arsitektur baru yang berbeda. Dengan menggunakan library Cypress saja sudah dapat melakukan testing sebuah frontend website. Dengan bentuk program yang hanya menggunakan JavaScript, Cypress berjalan dengan cepat, ditambah dengan fitur dashboard yang mempermudah untuk melakukan testing kepada suatu front-end.

BAB IV
IMPLEMENTASI SISTEM

Pada bab ini menjelaskan tahap implementasi yang dilakukan selama kerja praktik. Selama kerja praktik tiga bulan, sebenarnya saya membuat aplikasi CCTools secara menyeluruh bersama dua karyawan divisi Operational Management lainnya. Namun dalam buku kerja praktik ini, yang akan dijelaskan adalah terfokus pada fitur manage member & role, dan fitur login dari aplikasi CCTools. Pada tiap fitur, terdapat banyak fungsi yang digunakan, sehingga didapatkan aplikasi CCTools yang dapat berjalan sesuai kebutuhan.

4.1. Login ke aplikasi CCTools

4.1.1 Deskripsi

Membuat fungsi untuk login ke aplikasi CCTools menggunakan username dan password.

4.1.2. Parameter

Terdapat 2 parameter yang digunakan dari fungsi ini. Berikut adalah parameter dari fungsi login ke aplikasi CCTools

Tabel 4.1 Parameter fungsi login ke aplikasi CCTools

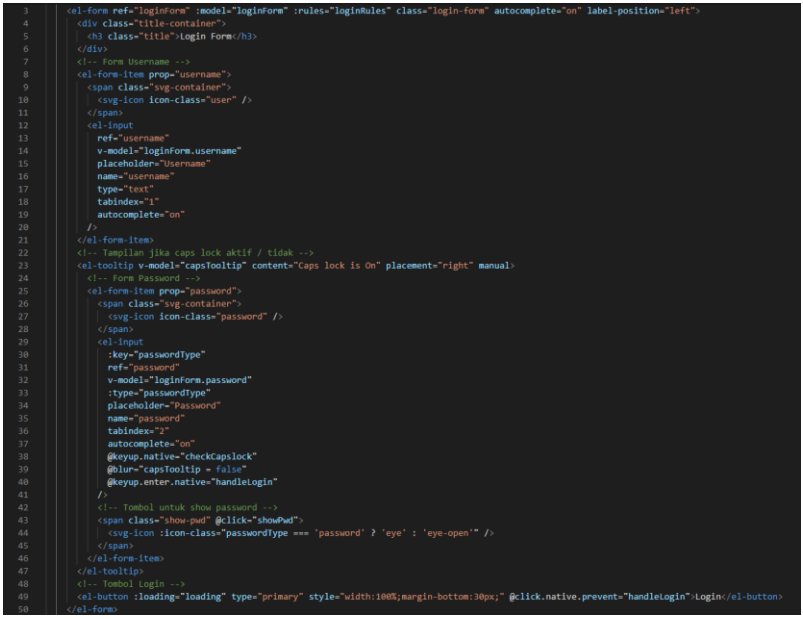
Nama Parameter	Keterangan
Username	Username dari pengguna, bersifat unik.
Password	Password dari pengguna, bersifat unik.

4.1.3. Data

Data yang digunakan untuk diolah dengan fungsi ini yaitu username dan password dari pengguna, jika benar maka state akun akan disimpan sebagai *key-login*, sedangkan jika salah maka akan terdapat peringatan bahwa username atau password salah.

4.1.4. Source Code

Pada fungsi login, terdapat source code pada frontend untuk memanggil data dari backend, dan terdapat source code backend untuk mengambil data dari database.



```
<el-form ref="loginForm" :model="loginForm" :rules="loginRules" class="login-form" autocomplete="on" label-position="left">
  <div class="title-container">
    <h3 class="title">Login Form</h3>
  </div>
  <!-- Form Username -->
  <el-form-item prop="username">
    <span class="svg-container">
      <svg-icon icon-class="user" />
    </span>
    <el-input
      ref="username"
      v-model="loginForm.username"
      placeholder="Username"
      name="username"
      type="text"
      tabindex="1"
      autocomplete="on"
    />
  </el-form-item>
  <!-- Tampilan jika caps lock aktif / tidak -->
  <el-tooltip v-model="capsTooltip" content="Caps lock is On" placement="right" manual>
    <!-- Form Password -->
    <el-form-item prop="password">
      <span class="svg-container">
        <svg-icon icon-class="password" />
      </span>
      <el-input
        :key="passwordType"
        ref="password"
        v-model="loginForm.password"
        :type="passwordType"
        placeholder="Password"
        name="password"
        tabindex="2"
        autocomplete="on"
        @keyup.native="checkCapslock"
        @blur="capsTooltip = false"
        @keyup.enter.native="handleLogin"
      />
      <!-- Tombol untuk show password -->
      <span class="show-pwd" @click="showPwd">
        <svg-icon :icon-class="passwordType === 'password' ? 'eye' : 'eye-open'" />
      </span>
    </el-form-item>
  </el-tooltip>
  <!-- Tombol Login -->
  <el-button :loading="loading" type="primary" style="width:100%;margin-bottom:30px;" @click.native.prevent="handleLogin">Login</el-button>
</el-form>
```

Kode 4.1 Tampilan Dashboard Login

```
methods: {
  //Check if Caps Lock is On
  checkCapslock(e) {
    const { key } = e
    this.capsTooltip = key && key.length === 1 && (key >= 'A' && key <= 'Z')
  },
  //Show Password
  showPwd() {
    if (this.passwordType === 'password') {
      this.passwordType = ''
    } else {
      this.passwordType = 'password'
    }
    this.$nextTick(() => {
      this.$refs.password.focus()
    })
  },
  //Login Handler
  handleLogin() {
    this.$refs.loginForm.validate(valid => {
      if (valid) {
        this.loading = true
        this.$store.dispatch('user/login', this.loginForm)
          .then(() => {
            this.$router.push({ path: this.redirect || '/', query: this.otherQuery })
            this.loading = false
          })
          .catch((error) => {
            console.log(error)
            this.loading = false
          })
      } else {
        console.log('error submit!!')
        return false
      }
    })
  },
```

Kode 4.2 Fungsi handleLogin, showPwd, dan checkCapslock

```
33    login({ commit }, userInfo) {
34      const { username, password } = userInfo
35
36      return new Promise((resolve, reject) => {
37        ccToolsLogin(username, password)
38          .then(res => {
39            if (!res.status) {
40              reject('Please, make sure you use correct user and password')
41            }
42
43            commit('SET_TOKEN', res.data.detail_role.name)
44            setToken(res.data.detail_role.name)
45            resolve()
46          })
47          .catch(error => {
48            reject(error)
49          })
50      })
51    },
```

Kode 4.3 Fungsi login pada store/user.js

```
13  export async function ccToolsLogin(username, password) {
14    const formData = new FormData()
15
16    formData.append('token', 'secret')
17    formData.append('action', 'api-validate-login-user')
18    formData.append('gateaway', 'user-helper')
19    formData.append('username', username)
20    formData.append('password', password)
21
22    const res = await axios.post(`${environment()}/ccTools/api/index`, formData, {})
23
24    return res.data
25  }
```

Kode 4.4 Fungsi request ke backend

```php
public function validateLogin($username, $password)
{
    try {
        $userModel                  = new User();
        $roleHelper                 = new RoleHelper();
        $execute                    = $userModel->validateUserLogin($username, $password);
        if ($execute['status'] != 1)
            throw new LogException($execute['message']);

        $dataResUser                = json_decode( json_encode($execute), true);
        $roleId                     = $dataResUser['data']['roleid'];
        $getRole                    =  $roleHelper->getById($roleId);
        if ($getRole['status'] != 1)
            throw new LogException($getRole['message']);

        $dataResRole                = json_decode( json_encode($getRole), true);
        $userData['detail_user']    = $dataResUser['data'];
        $userData['detail_role']    = $dataResRole['data'];
        $userData['message']        = 'Authentications is success';

        return [
            'status'    => 1,
            'data'      => $userData
        ];
    } catch (LogException $e) {
        return [
            'status'    => 0,
            'message'   => $e->getMessage()
        ];
    }
}
```

Kode 4.5 Untuk mengarah ke fungsi validateUserLogin

```php
public function validateUserLogin($username, $password){
    try {
        $query = DB::connection($this->connection)
            ->table($this->table)
            ->where('username', $username)
            ->where('password', $password)
            ->first();
        if (!$query)
            throw new LogException('Failed to validate user');

        return [
            'status'    => 1,
            'data'      => $query
        ];
    } catch (LogException $exception) {
        return [
            'status'    => 0,
            'message'   => $exception->getMessage(),
        ];
    }
}
```

Kode 4.6 Fungsi untuk request validasi login

Berdasarkan kode diatas, alur berawal dari user memasukkan form username dan password lalu menekan login. Kemudian, data username dan password dari form akan diarahkan ke backend, yang kemudian diarahkan ke request validasi username dan password, dimana jika username dan passwordnya sudah tepat, maka hasilnya adalah login berhasil, namun jika salah, akan terdapat warning bahwa username atau password salah.

4.2. Menampilkan list user

4.2.1 Deskripsi

Membuat fungsi untuk menampilkan list dari user yang dapat mengakses aplikasi CCTools.

4.2.2. Parameter

Tidak ada parameter yang digunakan pada fungsi ini.

4.2.3. Data

Data yang digunakan untuk fungsi ini adalah data dari user, mulai dari id, nama user, role user, dan password.

4.2.4. Source Code

Pada fungsi menampilkan list user, terdapat frontend sebagai dashboard tampilannya, dan backend untuk request query data dari user.

```
<el-table v-loading="loadList" :data="usersList" style="width: 100%;margin-top:30px;" border>
  <el-table-column align="center" label="id" width="220">
    <template slot-scope="{row}">
      {{ row.id }}
    </template>
  </el-table-column>
  <el-table-column align="center" label="Username" width="220">
    <template slot-scope="{row}">
      {{ row.username }}
    </template>
  </el-table-column>
  <el-table-column align="center" label="Email" min-width="300px">
    <template slot-scope="{row}">
      {{ row.email }}
    </template>
  </el-table-column>
  <el-table-column align="center" label="Role">
    <template slot-scope="{row}">
      {{ row.roleid }}
    </template>
  </el-table-column>
  <el-table-column align="center" label="Password" width="200px">
    <template slot-scope="{row}">
      {{ row.password }}
    </template>
  </el-table-column>
  <el-table-column align="center" label="Operations" width="250px">
    <template slot-scope="scope">
      <el-button type="primary" size="small" @click="handleEdit(scope)">Edit</el-button>
      <el-button type="danger" size="small" @click="handleDelete(scope)">Delete</el-button>
    </template>
  </el-table-column>
</el-table>
```

Kode 4.7 Tampilan dashboard list user

```
getUsers() {
  const formData = new FormData()
  formData.append('token', 'secret')
  formData.append('action', 'api-fetch-user')
  formData.append('gateaway', 'user-helper')
  axios.post(`${environment()}/ccTools/api/index`, formData, {}).then((res) => {
    if (res.data.status === 1) {
      this.usersList = res.data.data
      this.loadList = false
    } else {
      this.$notify({
        title: 'Failed',
        message: 'No data',
        type: 'warning'
      })
    }
  })
},
```

Kode 4.8 Fungsi getUsers untuk request ke backend

```php
public function fetch()
{
    try {
        $userModel  = new User();
        $execute    = $userModel->fetch();
        if ($execute['status'] != 1)
            throw new LogException($execute['message']);

        $userData       = json_decode( json_encode($execute), true);
        return [
            'status'    => 1,
            'data'      => $userData['data']
        ];
    } catch (LogException $e) {
        return [
            'status'    => 0,
            'message'   => $e->getMessage()
        ];
    }
}
```

Kode 4.9 Backend fetchUser untuk request query data user

```php
public function fetch(){
    try {
        $query = DB::connection($this->connection)
            ->table($this->table)
            ->get()->toArray();
        if (!$query)
            throw new LogException('Failed to fetch data');

        return [
            'status'    => 1,
            'data'      => $query
        ];
    } catch (LogException $exception) {
        return [
            'status'    => 0,
            'message'   => $exception->getMessage(),
        ];
    }
}
```

Kode 4.10 Query list user ke database

Berdasarkan kode di atas, alur dari fungsi list user adalah saat halaman diakses, maka fungsi getUsers() akan dijalankan untuk request ke backend. Dari backend, akan diarahkan ke fungsi query data list user ke database.

4.3. Menambahkan dan atau Mengubah User

4.3.1. Deskripsi

Membuat fungsi untuk menambahkan user baru dan atau mengubah data user yang sudah ada di aplikasi CCTools.

4.3.2. Parameter

Parameter dari fungsi ini adalah

Tabel 4.2 Parameter fungsi menambahkan dan atau mengubah user

Nama Parameter	Keterangan
Username	Username dari user
Password	Password dari user
RoleID	Role yang dimiliki untuk limit akses user
Email	Email dari user

4.3.3. Data

Data yang digunakan untuk diolah pada fungsi ini yaitu form add User yang akan di post ke database dari aplikasi CCTools. Pada edit User, data yang digunakan adalah data user yang telah ada, namun telah diubah melalui form edit user yang akan di put ke database aplikasi CCTools.

4.3.4. Source Code

Pada fungsi menambahkan user, terdapat tombol “Add” di dashboard halaman user, yang jika ditekan akan menampilkan form add user yang mana setelah memasukkan data user, akan dipost ke database aplikasi CCTools. Sedangkan pada edit user terdapat tombol “Edit” pada tabel list user di setiap sebelah kanan data user.

Kedua fungsi ini dijelaskan dalam satu bagian dikarenakan method yang digunakan tidak berbeda jauh antara keduanya. Hal ini dikarenakan Add User dan Edit User menggunakan format form yang sama. Bedanya, pada form edit user sudah terisi data user tersebut yang dapat diubah, sedangkan add user form masih kosong.

```
39    <el-dialog :visible.sync="dialogVisible" :title="dialogType==='edit'?'Edit User':'New User'">
40      <el-form :model="user" label-width="80px" label-position="left">
41        <el-form-item v-if="dialogType==='edit'" label="ID">
42          {{ user.id }}
43        </el-form-item>
44        <el-form-item label="Username">
45          <el-input v-model="user.username" />
46        </el-form-item>
47        <el-form-item label="Roles">
48          <el-select v-model="listQuery" placeholder="Roles" clearable class="filter-item">
49            <el-option
50              v-for="role in roleOptions"
51              :key="role.key"
52              :label="role.name"
53              :value="role.id"
54            />
55          </el-select>
56        </el-form-item>
57        <el-form-item label="E-Mail">
58          <el-input v-model="user.email" />
59        </el-form-item>
60        <el-form-item label="Password">
61          <el-input v-model="user.password" />
62        </el-form-item>
63      </el-form>
64      <div style="text-align:right;">
65        <el-button type="danger" @click="dialogVisible=false">Cancel</el-button>
66        <el-button type="primary" @click="confirmUser">Confirm</el-button>
67      </div>
68    </el-dialog>
```

Kode 4.11 Tampilan form saat menekan add / edit

```
169     handleAdd() {
170       this.user = Object.assign({}, defaultUser)
171       this.dialogType = 'new'
172       this.dialogVisible = true
173     },
```

Kode 4.12 Add Handler

```
174     handleEdit(scope) {
175       this.dialogType = 'edit'
176       this.dialogVisible = true
177       this.user = deepClone(scope.row)
178       this.listQuery = scope.row.roleid
179     },
```

Kode 4.13 Edit Handler

```
confirmUser() {
  const isEdit = this.dialogType === 'edit'
  const formData = new FormData()
  if (isEdit) {
    formData.append('token', 'secret')
    formData.append('action', 'api-update-user')
    formData.append('gateaway', 'user-helper')
    formData.append('username', this.user.username)
    formData.append('password', this.user.password)
    formData.append('roleid', this.listQuery)
    formData.append('email', this.user.email)
    formData.append('id', this.user.id)
    axios.post(`${environment()}/ccTools/api/index`, formData, {}).then((res) => {
      if (res.data.status === 1) {
        this.loadList = true
        this.getUsers()
        this.$notify({
          title: 'Success',
          message: res.data.message,
          type: 'success'
        })
      } else {
        this.$notify({
          title: 'Failed',
          message: res.data.message,
          type: 'warning'
        })
      }
    })
```

Kode 4.14 Fungsi Confirm apabila edit user

```
  } else {
    formData.append('token', 'secret')
    formData.append('action', 'api-add-user')
    formData.append('gateaway', 'user-helper')
    formData.append('username', this.user.username)
    formData.append('password', this.user.password)
    formData.append('roleid', this.listQuery)
    formData.append('email', this.user.email)
    axios.post(`${environment()}/ccTools/api/index`, formData, {}).then((res) => {
      if (res.data.status === 1) {
        this.loadList = true
        this.getUsers()
        this.$notify({
          title: 'Success',
          message: res.data.message,
          type: 'success'
        })
      } else {
        this.$notify({
          title: 'Failed',
          message: res.data.message,
          type: 'warning'
        })
      }
    })
  }
  this.dialogVisible = false
}
```

Kode 4.15 Fungsi Confirm apabila add user

```php
    public function addUser($username, $password, $email, $roleid)
    {
        try {
            $userModel  = new User();
            $params     = [
                'username'  => $username,
                'email'     => $email,
                'password'  => $password,
                'roleid'    => $roleid
            ];
            $execute        = $userModel->saveData($params);
            if ($execute['status'] != 1)
                throw new LogException($execute['message']);

            return [
                'status'    => 1,
                'message'   => $execute['message']
            ];
        } catch (LogException $e) {
            return [
                'status'    => 0,
                'message'   => $e->getMessage()
            ];
        }
    }
```

Kode 4.16 Backend untuk memanggil query post data

```php
public function updateUser($id, $username, $password, $email, $roleid)
{
    try {
        $userModel      = new User();
        $params         = [];
        $getUserData    = $this->getById($id);
        if ($getUserData['status'] != 1)
            throw new LogException($getUserData['message']);

        $userData       = json_decode( json_encode($getUserData), true);
        if (!empty($username)) {
            if ($userData['data']['username']   != $username) {
                $params['username']             = $username;
            }
        }

        if (!empty($roleid)) {
            if ($userData['data']['roleid']     != $roleid) {
                $params['roleid']               = $roleid;
            }
        }

        if (!empty($password)) {
            if (md5($userData['data']['password'])  != md5($password)) {
                $params['password']                 = $password;
            }
        }

        if (!empty($email)) {
            if (!filter_var($email, FILTER_VALIDATE_EMAIL)) {
                throw new LogException('Invalid email format');
            }

            if ($userData['data']['email']      != $email) {
                $params['email']                = $email;
            }
        }

        if (!empty(array_filter($params))) {
            $execute = $userModel->update($id, $params);
            if ($execute['status'] != 1)
                throw new LogException($execute['message']);
        }
        else {
            throw new LogException('Nothing to update');
        }

        return [
            'status'    => 1,
            'message'   => $execute['message']
        ];
    } catch (LogException $e) {
        return [
            'status'    => 0,
            'message'   => $e->getMessage()
        ];
    }
}
```

Kode 4.17 Backend untuk memanggil query update data

```php
public function saveData($params){
    try {
        $query = DB::connection($this->connection)
            ->table($this->table)
            ->insert($params);

        if (!$query)
            throw new \Exception('Failed to save data');

        return [
            'status'    => 1,
            'message'   => 'Success to save data'
        ];
    } catch (\Exception $exception) {
        return [
            'status'    => 0,
            'message'   => $exception->getMessage(),
        ];
    }
}
```

Kode 4.18 Fungsi Query post data

```php
public function update($id, $params){
    try {
        $query = DB::connection($this->connection)
            ->table($this->table)
            ->where('id', $id)
            ->update($params);
        if (!$query)
            throw new LogException('Failed to update data');

        return [
            'status'    => 1,
            'message'   => 'Success to update data'
        ];
    } catch (LogException $exception) {
        return [
            'status'    => 0,
            'message'   => $exception->getMessage(),
        ];
    }
}
```

Kode 4.19 Fungsi Query update data

Berdasarkan kode di atas, add dan edit user memiliki alur yang mirip, yaitu dimulai dari menekan tombol add / edit, lalu akan muncul form. Setelah mengisi data di form, tekan confirm yang akan menjalankan *confirmUser()*. Dari situ akan diarahkan ke backend, pada backend akan request fungsi query, dimana jika add user maka dia post data, sedangkan jika edit user maka dia update data.

4.4. Menghapus user

4.4.1. Deskripsi

Membuat fungsi untuk menghapus user pada aplikasi CCTools.

4.4.2. Parameter

Tidak ada parameter yang digunakan pada fungsi ini.

4.4.3. Data

Data yang digunakan pada fungsi ini adalah id dari user tersebut, yang kemudian akan dihapus dari database.

4.4.4. Source Code

Pada fungsi menghapus user, alurnya cukup simple. Yang perlu dilakukan adalah menekan tombol delete, yang selanjutnya akan menjalankan request delete user tertentu dari database.

```
handleDelete({ $index, row }) {
  this.user = deepClone(row)
  const formData = new FormData()
  formData.append('token', 'secret')
  formData.append('action', 'api-delete-user')
  formData.append('gateaway', 'user-helper')
  formData.append('id', this.user.id)
  axios.post(`${environment()}/ccTools/api/index`, formData, {}).then((res) => {
    if (res.data.status === 1) {
      this.loadList = true
      this.getUsers()
    } else {
      this.$notify({
        title: 'Failed',
        message: res.message,
        type: 'warning'
      })
    }
  })
},
```

Kode 4.20 Fungsi untuk handle delete user di frontend

```
    public function deleteUserById($id)
    {
        try {
            $userModel      = new User();
            $getUserData    = $this->getById($id);
            if ($getUserData['status'] != 1)
                throw new LogException($getUserData['message']);

            $execute        = $userModel->deleteById($id);
            if ($execute['status'] != 1)
                throw new LogException($execute['message']);

            return [
                'status'    => 1,
                'message'   => $execute['message']
            ];
        } catch (LogException $e) {
            return [
                'status'    => 0,
                'message'   => $e->getMessage()
            ];
        }
    }
```

Kode 4.21 Backend untuk diarahkan ke query delete user

```php
    public function deleteById($id){
        try {
            $query = DB::connection($this->connection)
                ->table($this->table)
                ->where('id', $id)
                ->delete();
            if (!$query)
                throw new LogException('Failed to delete data');

            return [
                'status'    => 1,
                'message'   => 'Success to delete data'
            ];
        } catch (LogException $exception) {
            return [
                'status'    => 0,
                'message'   => $exception->getMessage(),
            ];
        }
    }
```

Kode 4.22 Fungsi Query delete user berdasarkan id

4.5. Menampilkan list role

4.5.1. Deskripsi

Membuat fungsi untuk menampilkan list role pada halaman role di aplikasi CCTools.

4.5.2. Parameter

Tidak ada parameter yang digunakan pada fungsi ini.

4.5.3. Data

Data yang dipakai dalam menampilkan list role diantaranya adalah id, nama role, akses path, dan tanggal update role.

4.5.4. Source Code

Pada fungsi menampilkan list role, terdapat frontend sebagai dashboard tampilan, dan backend yang berguna untuk request get data dari tabel tbl_role.

```
<el-table
  :key="tableKey"
  v-loading="listLoading"
  :data="list"
  border
  fit
  highlight-current-row
  style="width: 100%;"
>
  <el-table-column label="ID" width="50px" align="center">
    <template slot-scope="{row}">
      <span>{{ row.id }}</span>
    </template>
  </el-table-column>
  <el-table-column label="Name" width="150px" align="center">
    <template slot-scope="{row}">
      <span>{{ row.name }}</span>
    </template>
  </el-table-column>
  <el-table-column label="Roles" min-width="363px" align="center">
    <template slot-scope="{row}">
      <span>{{ row.roles }}</span>
    </template>
  </el-table-column>
  <el-table-column label="Last Update" width="180px" align="center">
    <template slot-scope="{row}">
      <span>{{ row.updated_at }}</span>
    </template>
  </el-table-column>
  <el-table-column label="Actions" align="center" width="350" class-name="small-padding fixed-width">
    <template slot-scope="{row}">
      <el-button type="primary" size="mini" @click="handleUpdate(row)">
        Edit
      </el-button>
      <el-button size="mini" type="danger" @click="handleDelete(row)">
        Delete
      </el-button>
    </template>
  </el-table-column>
</el-table>
```

Kode 4.23 Tampilan tabel list user

```
getList() {
  this.listLoading = true
  const formData2 = new FormData()
  formData2.append('token', 'secret')
  formData2.append('action', 'api-fetch-role')
  formData2.append('gateaway', 'role-helper')
  axios.post(`${environment()}/ccTools/api/index`, formData2, {}).then((res) => {
    if (res.data.status === 1) {
      this.successMessageAction('Success to fetch data')
      this.list = res.data.data
      this.listLoading = false
    } else if (res.data.message === 'Failed to fetch data') {
      this.listLoading = false
      this.failMessageAction('Data is empty')
      return false
    } else {
      this.failMessageAction(res.data.message)
      this.listLoading = false
      return false
    }
  })
},
```

Kode 4.24 Fungsi get list di frontend

```php
public function fetch( $name = '' )
{
    try {
        $roleModel      = new Role();
        $execute        = $roleModel->fetch( $name );
        if ($execute['status'] != 1)
            throw new LogException($execute['message']);

        $userData       = json_decode( json_encode($execute), true);
        return [
            'status'    => 1,
            'data'      => $userData['data']
        ];
    } catch (LogException $e) {
        return [
            'status'    => 0,
            'message'   => $e->getMessage()
        ];
    }
}
```

Kode 4.25 Fungsi backend untuk request Query get data

```php
public function fetch($name){
    try {
        $query = DB::connection($this->connection)
            ->table($this->table);

        if ($name)
            $query = $query
                ->where('name','=',$name);

        $query = $query
            ->get()
            ->toArray();

        if (!$query)
            throw new LogException('Failed to fetch data');

        return [
            'status'    => 1,
            'data'      => $query
        ];
    } catch (LogException $exception) {
        return [
            'status'    => 0,
            'message'   => $exception->getMessage(),
        ];
    }
}
```

Kode 4.26 Fungsi Query get data role

4.6. Menambahkan dan atau Mengubah Role

4.6.1. Deskripsi

Membuat fungsi untuk menambahkan role baru dan atau mengubah data role yang sudah ada di aplikasi CCTools.

4.6.2. Parameter

Parameter dari fungsi ini adalah

Tabel 4.3 Parameter fungsi menambahkan dan atau mengubah role

Nama Parameter	Keterangan
Id role	Username dari user
Nama	Password dari user
Akses Path	Role yang dimiliki untuk limit akses user

4.6.3. Data

Data yang digunakan untuk diolah pada fungsi ini yaitu form add Role yang akan di post ke database dari aplikasi CCTools. Pada edit Role, data yang digunakan adalah data role yang telah ada, namun telah diubah melalui form edit role yang akan di put ke database aplikasi CCTools.

4.6.4. Source Code

Pada fungsi menambahkan role, terdapat tombol “Add” di dashboard halaman role, yang jika ditekan akan menampilkan form add role yang mana setelah memasukkan data role, akan dipost ke database aplikasi CCTools. Sedangkan pada edit role terdapat tombol “Edit” pada tabel list role di setiap sebelah kanan data role.

Kedua fungsi ini dijelaskan dalam satu bagian dikarenakan method yang digunakan tidak berbeda jauh antara keduanya. Hal ini dikarenakan Add dan Edit role menggunakan format form yang sama. Bedanya, pada form edit role sudah terisi data role sesuai id nya tersebut yang dapat diubah, sedangkan add role form masih kosong.

```
<el-dialog :title="textMap[dialogStatus]" :visible.sync="formAdd">
  <el-form
    ref="dataFormAdd"
    :model="temp"
    label-position="left"
    label-width="70px"
    style="width: 400px; margin-left:50px;"
  >
    <el-form-item label="Name" prop="name">
      <el-input v-model="temp.name" />
    </el-form-item>
    <el-form-item label="Roles" prop="role">
      <el-tree
        ref="tree"
        :data="routes"
        :props="defaultProps"
        show-checkbox
        :default-checked-keys="[]"
        node-key="path"
        class="permission-tree"
      />
    </el-form-item>
  </el-form>
  <div slot="footer" class="dialog-footer">
    <el-button @click="cancel();formAdd = false">
      Cancel
    </el-button>
    <el-button type="primary" @click="dialogStatus===createData()">
      Confirm
    </el-button>
  </div>
</el-dialog>
```

Kode 4.27 Tampilan form add / edit role

```
cancel() {
  this.$refs.tree.setCheckedKeys([])
},
handleCheckChange() {
  const temp = this.$refs.tree.getCheckedKeys()
  this.$refs.tree.setCheckedKeys([])
  return temp
},
getRoutes() {
  const res = deepClone([...asyncRoutes])
  this.routes = this.generateRoutes(res)
},
generateRoutes(routes, basePath = '/') {
  const res = []

  for (let route of routes) {
    // skip some route
    if (route.hidden) { continue }

    const onlyOneShowingChild = this.onlyOneShowingChild(route.children, route)

    if (route.children && onlyOneShowingChild && !route.alwaysShow) {
      route = onlyOneShowingChild
    }

    const data = {
      path: path.resolve(basePath, route.path),
      title: route.meta && route.meta.title

    }

    // recursive child routes
    if (route.children) {
      data.children = this.generateRoutes(route.children, data.path)
    }
    res.push(data)
  }
  return res
},
```

Kode 4.28 Generate akses path pada form role

```
273     handleCreate() {
274       this.resetTemp()
275       this.dialogStatus = 'create'
276       this.formAdd = true
277       this.$nextTick(() => {
278         this.$refs['dataFormAdd'].clearValidate()
279       })
280     },
281     createData() {
282       var res = JSON.stringify(this.handleCheckChange())
283       this.$refs['dataFormAdd'].validate((valid) => {
284         if (valid) {
285           this.listLoading = true
286           const formData = new FormData()
287           formData.append('token', 'secret')
288           formData.append('action', 'api-add-role')
289           formData.append('gateaway', 'role-helper')
290           formData.append('name', this.temp.name)
291           formData.append('roles', res)
292           axios.post(`${environment()}/ccTools/api/index`, formData, {}).then((res) => {
293             if (res.data.status === 1) {
294               this.list = res.data.data
295               this.formAdd = false
296               this.successMessageAction(res.data.message)
297               this.getList()
298             } else {
299               this.failMessageAction(res.data.message)
300               this.formAdd = false
301               this.listLoading = false
302               return false
303             }
304           })
305         }
306       })
307     },
```

Kode 4.29 Create Handler di frontend

```
handleUpdate(row) {
  if (row.roles) {
    this.listRole = JSON.parse(row.roles)
  }
  this.temp = Object.assign({}, row) // copy obj
  this.dialogStatus = 'update'
  this.formUpdate = true
  this.$nextTick(() => {
    this.$refs['dataFormUpdate'].clearValidate()
  })
},
updateData() {
  var res = JSON.stringify(this.handleCheckChange())
  this.$refs['dataFormUpdate'].validate((valid) => {
    if (valid) {
      if (valid) {
        this.listLoading = true
        const formData = new FormData()
        formData.append('token', 'secret')
        formData.append('action', 'api-update-role')
        formData.append('gateaway', 'role-helper')
        formData.append('name', this.temp.name)
        formData.append('roles', res)
        formData.append('id', this.temp.id)
        axios.post(`${environment()}/ccTools/api/index`, formData, {}).then((res) => {
          if (res.data.status === 1) {
            this.list = res.data.data
            this.formUpdate = false
            this.successMessageAction(res.data.message)
            this.getList()
          } else {
            this.failMessageAction(res.data.message)
            this.formUpdate = false
            this.listLoading = false
            return false
          }
        })
      }
    }
  })
},
```

Kode 4.30 Update Handler di frontend

```php
public function addRole($name, $roles)
{
    try {
        $rolesModel = new Role();
        $params     = [
            'name'      => $name,
            'roles'     => $roles
        ];
        $execute        = $rolesModel->saveData($params);
        if ($execute['status'] != 1)
            throw new LogException($execute['message']);

        return [
            'status'    => 1,
            'message'   => $execute['message']
        ];
    } catch (LogException $e) {
        return [
            'status'    => 0,
            'message'   => $e->getMessage()
        ];
    }
}
```

Kode 4.31 Add Role di backend untuk request query post

```php
public function updateRole($id, $name, $roles)
{
    try {
        $roleModel      = new Role();
        $params         = [];
        $execute        = null;
        $getRoleData    = $this->getById($id);
        if ($getRoleData['status'] != 1)
            throw new LogException($getRoleData['message']);

        $roleData       = json_decode( json_encode($getRoleData), true);
        if (!empty($name)) {
            if ($roleData['data']['name']   != $name) {
                $params['name']             = $name;
            }
        }

        if (!empty($roles)) {
            if ($roleData['data']['roles']  != $roles) {
                $params['roles']            = $roles;
            }
        }

        if (!empty(array_filter($params))) {
            $execute = $roleModel->update($id, $params);
            if ($execute['status'] != 1)
                throw new LogException($execute['message']);
        }
        else {
            throw new LogException('Nothing to update');
        }

        return [
            'status'    => 1,
            'message'   => $execute['message']
        ];
    } catch (LogException $e) {
        return [
            'status'    => 0,
            'message'   => $e->getMessage()
        ];
    }
}
```

Kode 4.32 Edit role di backend untuk request query update

```php
    public function saveData($params){
        try {
            $query = DB::connection($this->connection)
                ->table($this->table)
                ->insert($params);

            if (!$query)
                throw new \Exception('Failed to save data');

            return [
                'status'    => 1,
                'message'   => 'Success to save data'
            ];
        } catch (\Exception $exception) {
            return [
                'status'    => 0,
                'message'   => $exception->getMessage(),
            ];
        }
    }
```

Kode 4.33 Query post data role

```php
    public function update($id, $params){
        try {
            $query = DB::connection($this->connection)
                ->table($this->table)
                ->where('id', $id)
                ->update($params);
            if (!$query)
                throw new LogException('Failed to update data');

            return [
                'status'    => 1,
                'message'   => 'Success to update data'
            ];
        } catch (LogException $exception) {
            return [
                'status'    => 0,
                'message'   => $exception->getMessage(),
            ];
        }
    }
```

Kode 4.34 Query update data role berdasarkan id

Berdasarkan kode di atas, alur dari add dan edit role hamper sama, dimana saat menekan tombol add / edit, akan tampil form, yang selanjutnya mengisi data di form. Pada form terdapat tampilan checkbox akses path yang di generate menggunakan fungsi Generate akses path pada Kode 4.28. Hal ini bertujuan mempermudah user untuk memilih path / halaman mana saja yang dapat diakses oleh suatu role. Hasil data dari Kode 4.28 berupa teks yang berisi semua path yang dipilih pada checkbox. Setelah mengisi form dan memilih akses path, dijalankan fungsi yang mengarah ke backend. Di backend, data dari form (termasuk akses path) akan di request menggunakan query.

4.7. Menghapus role

4.7.1. Deskripsi

Membuat fungsi untuk menghapus role pada aplikasi CCTools.

4.7.2. Parameter

Tidak ada parameter yang digunakan pada fungsi ini.

4.7.3. Data

Data yang digunakan pada fungsi ini adalah id dari role tersebut, yang kemudian akan dihapus dari database.

4.7.4. Source Code

Pada fungsi menghapus role, alurnya cukup simple. Yang perlu dilakukan adalah menekan tombol delete. Tombol delete terletak di sebelah kanan setiap list role, bersama dengan tombol edit, sehingga saat menekan tombol delete dari suatu role, maka role tersebut akan dihapus dari database.

```
handleDelete(row) {
  this.listLoading = true
  const formData = new FormData()
  formData.append('token', 'secret')
  formData.append('action', 'api-delete-role')
  formData.append('gateaway', 'role-helper')
  formData.append('id', row.id)
  axios.post(`${environment()}/ccTools/api/index`, formData, {}).then((res) => {
    if (res.data.status === 1) {
      this.list = res.data.data
      this.formUpdate = false
      this.successMessageAction(res.data.message)
      this.getList()
    } else {
      this.failMessageAction(res.data.message)
      this.formUpdate = false
      this.listLoading = false
      return false
    }
  })
}
```

Kode 4.35 Fungsi Delete di frontend

```
    public function deleteRoleById($id)
    {
        try {
            $roleModel      = new Role();
            $getRoleData    = $this->getById($id);
            if ($getRoleData['status'] != 1)
                throw new LogException($getRoleData['message']);

            $execute        = $roleModel->deleteById($id);
            if ($execute['status'] != 1)
                throw new LogException($execute['message']);

            return [
                'status'    => 1,
                'message'   => $execute['message']
            ];
        } catch (LogException $e) {
            return [
                'status'    => 0,
                'message'   => $e->getMessage()
            ];
        }
    }
```

Kode 4.36 Delete di backend untuk request query delete

```php
    public function deleteById($id){
        try {
            $query = DB::connection($this->connection)
                ->table($this->table)
                ->where('id', $id)
                ->delete();
            if (!$query)
                throw new LogException('Failed to delete data');

            return [
                'status'    => 1,
                'message'   => 'Success to delete data'
            ];
        } catch (LogException $exception) {
            return [
                'status'    => 0,
                'message'   => $exception->getMessage(),
            ];
        }
    }
```

Kode 4.37 Query delete role berdasarkan id

BAB V
PENGUJIAN DAN EVALUASI

Bab ini menjelaskan tahap uji coba dilakukan terhadap fungsi yang telah dibuat.

5.1. Tujuan Pengujian

Pengujian dilakukan untuk menentukan apakah fungsi tersebut telah memenuhi kriteria fungsi yang diminta oleh pengguna dan dapat berfungsi sebagaimana mestinya yang dibutuhkan.

5.2. Kriteria Pengujian

Penilaian atas pencapaian tujuan pengujian didapatkan dengan memperhatikan beberapa hasil yang diharapkan berikut ini:

a. Kesesuaian dengan kriteria fungsi yang diminta oleh pengguna

5.3. Skenario Pengujian

Pengujian dilakukan oleh pembimbing kerja praktik, dengan menggunakan dokumen UAT (User Acceptance Testing) sebagai acuan tiap fungsi yang telah dibuat.

5.4. Hasil Pengujian

Berdasarkan hasil kesepakatan dengan perusahaan, saya tidak diperbolehkan menyebar dokumen / data yang dimiliki oleh perusahaan, sehingga saya tidak dapat memperlihatkan bukti hasil pengujian di buku ini.

5.5. Evaluasi Pengujian

Hasil evaluasi pengujian fungsi query dapat dilihat pada tabel 5.1.

Tabel 5.1 Hasil Evaluasi Pengujian Fungsi

No.	Kriteria Pengujian	Hasil Pengujian
1	Login ke aplikasi CCTools	Terpenuhi
2	Menampilkan list user	Terpenuhi
3	Menambahkan dan atau mengubah user	Terpenuhi
4	Menghapus user	Terpenuhi
5	Menampilkan list role	Terpenuhi
6	Menambahkan dan atau mengubah role	Terpenuhi
7	Menghapus role	Terpenuhi

Dengan hasil pengujian pada tabel di atas, dapat disimpulkan bahwa secara keseluruhan fungsi-fungsi tersebut telah memenuhi kebutuhan dari karyawan PT. Beon Intermedia divisi Customer Care.

BAB VI
KESIMPULAN DAN SARAN

6.1. Kesimpulan

Kesimpulan yang didapat setelah melakukan kegiatan kerja praktik di PT. Beon Intermedia adalah sebagai berikut:

- Dengan adanya aplikasi CCTools, para karyawan menjadi lebih mudah untuk memonitoring pelanggan dari PT. Beon Intermedia.
- Framework VueJS sangat berguna dalam membuat sebuah frontend yang lebih baik dan lebih indah.
- Framework Phalcon membantu dalam membuat backend yang aman dan lebih terstruktur.

6.2. Saran

Saran untuk pengembangan sistem aplikasi CCTools pada fitur manage member dan loginnya adalah sebagai berikut:

- Pada fungsi list user dilakukan perubahan berupa encoding pada password agar password yang ditampilkan di table bukanlah password asli, melainkan yang sudah di encode.
- Pada fitur login, dapat dikembangkan berupa dibuatnya limit waktu tiap user untuk login, agar jika sudah melebihi waktu yang ditentukan, user akan otomatis logout.
- Pada fitur login, dapat dikembangkan dengan dibatasi satu user untuk login pada 1 perangkat saja, sehingga tidak terdapat user sama yang mengakses aplikasi di 2 perangkat berbeda (hal ini juga untuk mengurangi resiko pembobolan).

[Halaman ini sengaja dikosongkan]

DAFTAR PUSTAKA

[1] Beon, “Tentang Beon.co.id Unlimited Web Hosting Indonesia,”. [Online]
Available
https://beon.co.id/about-us

[2] Wikipedia, “HTML,”. [Online]
Available
https://en.wikipedia.org/wiki/HTML

[3] W3schools, “HTML Introduction,”. [Online]
Available
https://www.w3schools.com/html/html_intro.asp

[4] Javascript, “About Javascript.com”. [Online]
Available
https://www.javascript.com/about

[5] Wikipedia, “JavaScript,”. [Online]
Available
https://en.wikipedia.org/wiki/JavaScript

[6] Wikipedia, “MySQL,”. [Online]
Available
https://en.wikipedia.org/wiki/MySQL

[7] MySQL, “Why MySQL?”. [Online]
Available
https://www.mysql.com/why-mysql/

[8] Visual Studio Code, “Learn to code with Visual Studio Code,”. [Online]
Available
https://code.visualstudio.com/learn

[9] Node.js, “About Node.js,”. [Online]
Available
https://nodejs.org/en/about/

[10] Vue.js, “Vue.js Introduction,”. [Online]
Available
https://vuejs.org/v2/guide/
[11] Phalcon, “Phalcon Documentation,”. [Online]
Available
https://docs.phalcon.io/4.0/en/introduction
[12] Cypress, “Testing has been broken for too long,”. [Online]
Available
https://www.cypress.io/how-it-works

BIODATA PENULIS I

Nama	: Ahmad Yahya Abdul Aziz
Tempat, Tanggal Lahir	: Malang, 1 Juli 2020
Jenis Kelamin	: Laki-laki
Agama	: Islam
Status	: Belum Menikah
Alamat Asal	: Perum. Srigading Dalam Kav. 20, Malang
Alamat Surabaya	: JL. Krukah Selatan Gang XIVA/7, Surabaya
Telepon	: 085236156392
Email	: ayahyaa17@gmail.com

PENDIDIKAN FORMAL

2017 – sekarang	: Mahasiswa S1 Informatika ITS
2014 – 2017	: MAN 3 Malang
2011 – 2014	: MTs Surya Buana Malang
2005 – 2011	: SD Islam Sabilillah Malang

KEMAMPUAN

- *Web Programming* (HTML, CSS, JS, PHP)
- *Programming* (C, C++, Go, Python)
- *Database Manajemen* (Oracle, MySQL)
- *Software* Perkantoran (Microsoft Word, Excel, PowerPoint)
- Bahasa (Indonesia, Inggris, Arab)

AKADEMIS

Kuliah	: Departemen Informatika, Fakultas Teknologi Elektro dan Informatika Cerdas, Institut Teknologi Sepuluh Nopember Surabaya
Angkatan	: 2017
Semester	: 7 (Tujuh)
IPK	: 3.93 (Semester 6)